SYSTEM AND METHOD FOR SIMULATION AND MODELING OF BATCH PROCESS MANUFACTURING FACILITIES USING PROCESS TIME LINES

Inventor: Peter Brown

Cross-Reference to Related Application

This application is a continuation-in-part application of and claims priority to U.S. Patent Application No. 09/019,777, filed February 6, 1998, which claims priority to U.S. Provisional Patent Application No. 60/037,387, filed February 7, 1997, the contents of both of which are incorporated herein by reference in their entirety.

Background of the Invention

Field of the Invention

The present invention relates generally to the design of large scale batch manufacturing facilities, and specifically to the design of biopharmaceutical drug manufacturing processes.

Related Art

Biopharmaceutical plants produce biopharmaceutical products through biological methods. Typical biopharmaceutical synthesis methods are mammalian cell culture, microbial fermentation and insect cell culture. Occasionally, biopharmaceutical products are produced from natural animal or plant sources or by a synthetic technique called solid phase synthesis. Mammalian cell culture, microbial fermentation and insect cell culture involve the growth of living cells and the extraction of biopharmaceutical products from the cells or the medium surrounding the cells. Solid phase synthesis and crude

tissue extraction are processes by which biopharmaceuticals are synthesized from chemicals or extracted from natural plant or animal tissues, respectively.

The process for producing biopharmaceuticals is complex. In addition to basic synthesis, additional processing steps of separation, purification, conditioning and formulation are required to produce the end product biopharmaceutical. Each of these processing steps includes additional unit operations. For example, the step of purification may include the step of Product Adsorption Chromatography, which may further include the unit operations of High Pressure Liquid Chromatography (HPLC), Medium Pressure Liquid Chromatography (MPLC), Low Pressure Liquid Chromatography (LPLC), etc. The production of biopharmaceuticals is complex because of the number, complexity and combinations of synthesis methods and processing steps possible. Consequently, the design of a biopharmaceutical plant is expensive.

Tens of millions of dollars can be misspent during the design and construction phases of biopharmaceutical plants due to inadequacies in the design process. Errors and inefficiencies are introduced in the initial design of the biopharmaceutical production process because no effective tools for modeling and simulating a biopharmaceutical production process exists. The inadequacies in the initial process design carry through to all phases of the biopharmaceutical plant design and construction. Errors in the basic production process design propagate through all of the design and construction phases, resulting in increased cost due to change orders late in the facility development project. For example, detailed piping and instrumentation diagrams (P&IDs) normally cost thousands of dollars per diagram. Problems in the biopharmaceutical production process design frequently necessitate the re-working of these detailed P&IDs. This adds substantially to the overall cost of design and construction of a biopharmaceutical plant.

There are generally three phases of biopharmaceutical plants which coincide with the different levels of drug approval by the FDA. A Clinical Phase I/II biopharmaceutical plant produces enough biopharmaceutical product to

support both phase I and phase II clinical testing of the product which may involve up to a few hundred patients. A Clinical Phase III biopharmaceutical plant produces enough biopharmaceutical product to support two to three-thousand patients during phase III clinical testing. A Clinical Phase III plant will also produce enough of the biopharmaceutical drug to support an initial commercial offering upon the licensing of the drug by the FDA for commercial sale. The successive phases represent successively larger biopharmaceutical facilities to support full scale commercial production after product licensing. Often the production process design is repeated for each phase, resulting in increased costs to each phase of plant development.

The design, architecture and engineering of biopharmaceutical plants is a several hundred million dollars per year industry because of the complex nature of biopharmaceutical production. Design of biopharmaceutical plants occurs in discrete phases. The first phase is the conceptual design phase. The first step in the conceptual design phase is identifying the high-level steps of the process that will produce the desired biopharmaceutical. Examples of high-level steps are synthesis, separation, purification and conditioning. After the high-level process steps have been identified, the unit operations associated with each of the high-level steps are identified. Unit operations are discrete process steps that make up the high-level process steps. In a microbial fermentation process, for example, the high-level step of synthesis may include the unit operations of inoculum preparation, flask growth, seed fermentation and production fermentation.

The unit operation level production process is typically designed by hand and is prone to errors and inefficiencies. Often, in the conceptual design phase, the specifications for the final production process are not complete. Therefore some of the equipment design parameters, unit operation yields and actual production rates for the various unit operations must be estimated. These factors introduce errors into the initial design base of the production process. Additionally, since the production process is designed by hand, attempting to

optimize the process for efficiency and production of biopharmaceutical products is impractically time consuming.

Scale calculations for each of the unit operations are performed to determine the size and capacity of the equipment necessary to produce the desired amount of product per batch. Included in the scale calculations is the number of batches per year needed to produce the required amount of biopharmaceutical product. A batch is a single run of the biopharmaceutical process that produces the product. Increasing the size and capacity of the equipment increases the amount of product produced per batch. The batch cycle time is the amount of time required to produce one batch of product. The amount of product produced in a given amount of time, therefore, is dependent upon the amount produced per batch, and the batch cycle time. The scale calculations are usually executed by hand to determine the size and capacity of the equipment that will be required in each of the unit operations. Since the scale calculations are developed from the original conceptual design parameters, they are also subject to the same errors inherent in the initial conceptual design base.

Typically a process flow diagram is generated after the scale calculations for the unit operations have been performed. The process flow diagram graphically illustrates the process equipment such as tanks and pumps necessary to accommodate the process for a given batch scale. The process flow diagram illustrates the different streams of product and materials through the different unit operations. Generally associated with the process flow diagram is a material balance table which shows the quantities of materials consumed and produced in each step of the biopharmaceutical production process. The material balance table typically includes rate information of consumption of raw materials and production of product. The process flow diagram and material balance table provides much of the information necessary to develop a preliminary equipment list. The preliminary equipment list shows the equipment necessary to carry out all of the unit operations in the manufacturing procedure. Since the process flow diagram, material balance table and preliminary equipment list are determined

from the original conceptual design parameters, they are subject to the same errors inherent in the initial conceptual design base.

A preliminary facility layout for the plant is developed from the process flow diagram, material balance table and preliminary equipment list. The preliminary facility layout usually begins with a bubble or block diagram of the plant that illustrates the adjacencies of rooms housing different high-level steps, as well as a space program which dimensions out the space and square footage of the building. From this information a preliminary equipment layout for the plant is prepared. The preliminary equipment layout attempts to show all the rooms in the plant, including corridors, staircases, etc. Mechanical, electrical and plumbing engineers estimate the mechanical, electrical and plumbing needs, respectively, of the facility based on the facility design layout and the utility requirements of the manufacturing equipment. Since the preliminary facility layout is developed from the original conceptual design parameters, they are subject to the same errors inherent in the initial conceptual design base.

Typically the next phase of biopharmaceutical plant design is preliminary piping and instrumentation diagram (P&ID) design. Preliminary P&IDs are based on the process flow diagram from the conceptual design phase. Often the calculations on the process design are re-run and incorporated into the preliminary P&ID. The preliminary P&IDs incorporate the information from the material balance table with the preliminary equipment list to show the basic piping and instrumentation required to run the manufacturing process.

Detailed design is the next phase of biopharmaceutical plant design. Plans and specifications which allow vendors and contractors to bid on portions of the biopharmaceutical plant are developed during the detailed design. Detailed P&IDs are developed which schematically represent every detail of the process systems for the biopharmaceutical plant. The detailed P&IDs include for example, the size and components of process piping, mechanical, electrical and plumbing systems; all tanks, instrumentation, controls and hardware. A bill of materials and detailed specification sheets on all of the equipment and systems

are developed from the P&IDs. Detailed facility architecture diagrams are developed that coincide with the detailed P&IDs and equipment specifications. The detailed P&IDs and facility construction diagrams allow builders and engineering companies to bid on the biopharmaceutical plant project. Since the preliminary and detailed P&IDs are developed from the original conceptual design parameters, they are subject to the same errors inherent in the initial conceptual design base. Reworking the preliminary and detailed P&IDs due to errors in the conceptual design phase can cost thousands of dollars per diagram.

The inability to accurately model and simulate the biopharmaceutical production process drives inaccurate initial design. Often, these inaccuracies result in changes to the design and construction diagrams at the plant construction site, or repair and reconstruction of the plant during the construction phase resulting in millions of dollars in additional cost.

What is needed, therefore, is a system and method for accurately simulating and modeling a biopharmaceutical production process. A method and system for simulating and modeling biopharmaceutical production process would allow designers to reduce the number of errors introduced into plant design at the earliest stages. Such a system and method would allow an engineer to validate the production process design and maximize the efficiency of the plant by finding the optimum equipment configurations. Such a system and method would generate detailed specifications for the equipment and process steps that would smooth the transition throughout all of the design phases and fix the cost of design and construction of a biopharmaceutical facility. The present invention can also be used for determining the cost of goods for a product.

Summary of the Invention

The present invention satisfies the above-stated needs by providing a method and system for simulating and modeling of batch process manufacturing facilities using process time lines. The method includes the steps of identifying

a production process sequence, the production process sequence including a plurality of subprocesses. At least one of the plurality of subprocesses includes a plurality of batch cycles, each of which includes a plurality of unit operations. Each of the unit operations are identified by unit operation identifiers. Next, the system and method retrieves the process parameter information from a master list for each of the unit operation identifiers in the process sequence. The process parameter information includes information on discrete tasks associated with each unit operation. After the steps of identifying and retrieving, the system and method generates a process schedule that identifies initiation and completion times for each of the discrete tasks in the process sequence. Next, a process time line using the operational parameters, the block flow diagram, the set of scheduling cycles for each of the sequence of unit operations is generated. The process time line is used as a tool for batch processing and facility design.

Brief Description of the Figures

The features and advantages of the present invention will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference numbers indicate identical or functionally similar elements. Additionally, the left-most digit of a reference number identifies the drawing in which the reference number first appears.

FIG. 1 illustrates a flow diagram of the process to generate a block flow diagram and a process time line according to the present invention.

FIG. 2 illustrates a flow diagram of the process for determining the necessary reactor volume according to the present invention.

FIG. 3 illustrates a unit operation list for a microbial fermentation process.

FIG. 4 illustrates a unit operation list for a mammalian cell culture process.

FIG. 5 illustrates a file and process flow diagram for cross-referencing a unit operation list with a process parameters table according to the present

5   invention.

FIG. 6 illustrates an exemplary process parameters table.

FIG. 7 illustrates the process for generating a block flow diagram according to the present invention.

FIG. 8 illustrates an exemplary block flow diagram according to the

10   present invention.

FIG. 9 illustrates a block flow diagram for the process of generating a process time line according to the present invention.

FIGs. 10-11 illustrate a high-level process time line according to the present invention.

15   FIGs. 12A-12H illustrate a detailed process time line according to the present invention.

FIG. 13 illustrates an exemplary computer according to an embodiment of the present invention.

FIGs. 14-20 are detailed examples of a process parameters table showing

20   a list of unit operations and their associated parameters.

FIG. 21 illustrates a refined unit operation list for a mammalian cell culture process according to an embodiment of the present invention.

FIG. 22 illustrates a refined unit operation list for a microbial fermentation process according to an embodiment of the present invention.

5

FIG. 23A-F illustrate a refined process time line for a mammalian cell culture process according an embodiment of the present invention.

FIG. 24 is a flow chart that illustrates the method for determining the impact of design cycle offsets on the current critical path start time for a respective unit operation cycle.

10

FIG. 25 is a flow chart that illustrates the test for a new batch cycle according to an embodiment of the present invention.

FIG. 26 is a flow chart that illustrates the test for a new unit operation cluster cycle according to an embodiment of the present invention.

FIG. 27 is a flow chart that illustrates the test for a new unit operation

15

cycle according to an embodiment of the present invention.

FIGs. 28-34 is a flow chart that illustrates the determination of start/stop times for various cycles according to an embodiment of the present invention.

Detailed Description of the Preferred Embodiments

The preferred embodiment of the present invention is a computer based

20

system and method for the simulation and modeling of batch process manufacturing facilities. The preferred embodiment is based on a method for

generating scheduling information which accurately defines the complex manufacturing operations of batch manufacturing processes. This scheduling capability system allows the definition of manufacturing costs in a more detailed and accurate manner than previously possible. As a result, this invention allows the rapid and accurate evaluation of numerous batch manufacturing alternatives in order to arrive at an optimal process design early in a facility development project. In so doing the invention minimizes project cost over runs which result from inaccuracies that can carry forward from the early stages of design into construction. The invention also defines operations scheduling in a manner that improves the accuracy of material resource planning and preventative maintenance programs for operating manufacturing plants.

FIG. 1 illustrates a high-level flow diagram of the preferred embodiment. The process begins by determining the necessary reactor vessel capacity at step 102. The reactor vessel is the container in which the crude product is first synthesized. For example, in mammalian cell culture processes, the reactor vessel houses the mammalian cells suspended in growth media. Next, the unit operation sequence for production of the biopharmaceutical product is determined at step 104. The unit operation sequence is the series of unit operations that are required to produce the biopharmaceutical product. Each unit operation is an individual step in the biopharmaceutical manufacturing process with an associated set of manufacturing equipment. The unit operation list is the list of unit operations that make up the unit operation sequence and their associated sequence information. The unit operation sequence information is the information that defines the scheduling cycles for each of the unit operations in the unit operation list. Scheduling cycles are iterations of unit operations in the unit operation sequence. Together, the unit operation list and the unit operation sequence information define the unit operation sequence. The desired biopharmaceutical product dictates the particular unit operations and their order in the biopharmaceutical production process. Some examples of unit operations

are: inoculum preparation, initial seeding of the reactor vessel, solids harvest by centrifugation, high-pressure homogenization, dilution, etc.

Scheduling cycles and cycle offset duration for each of the unit operations in the biopharmaceutical production process are determined at step 106. Scheduling cycles are iterations of unit operations (the default being one (1)) in the unit operation sequence, and occur in three levels. Additionally, each level of scheduling cycle has an associated offset duration that dictates the time period between the beginnings of successive scheduling cycles.

"Cycles per Unit Operation" or preferably, "Unit Operation Cycles" (UC) is the first level of scheduling cycles. Unit Operation Cycles are defined as the number of iterations a unit operation is repeated in a process by itself before proceeding to the next unit operation. For example, the harvest and feed unit operation in a mammalian cell culture process has multiple Unit Operation Cycles. Product-rich media is drawn from the reactor vessel and nutrient-rich media is fed into the reactor vessel multiple times during one harvest and feed unit operation. The multiple draws of product-rich reactor media are pooled for processing in the next unit operation.

The second level of scheduling cycles is "Cycles per Batch" or preferably, "Unit Operation Cluster Cycles" (CC). Unit Operation Cluster Cycles are defined as the number of iterations a set of consecutive unit operations are repeated as a group before proceeding to the next unit operation after the set of consecutive unit operations. The set of consecutive unit operations repeated as a group are also referred to as a subprocess. For example, the set of unit operations including inoculum preparation, flask growth, seed fermentation, production fermentation, heat exchange, and continuous centrifugation/whole-cell harvest in a microbial fermentation process are often cycled together. Running through each of the six steps results in a single harvest from the microbial fermentation reactor vessel. Multiple harvests from a reactor vessel may be needed to achieve a batch of sufficient quantity. Each additional harvest is pooled with the previous harvest, resulting in a single batch of cell culture for the process.

The third level of scheduling cycles is "Cycles per Process" or preferably, "Batch Cycles" (BC). Batch Cycles are defined as the number of iterations a batch cycle is repeated for a process that employs continuous or semi-continuous product synthesis. In such a case, a single biopharmaceutical production process may result in multiple batches of product. For example, in a mammalian cell-culture process a single cell culture is typically in continuous production for 60-90 days. During this period multiple harvests of crude product are collected and pooled on a batch basis to be processed into the end product biopharmaceutical. The pooling of multiple harvests into a batch of material will occur several times during the cell culture period resulting in multiple Batch Cycles.

In step 108, a process parameters table master list is referenced to obtain all operational parameters for each unit operation in the unit operation list. The process parameters table contains a list of all unit operations and operational parameters necessary to simulate a particular unit operation. Examples of operational parameters are the solutions involved in a particular unit operation, temperature, pressure, duration, agitation, scaling volume, etc. Additionally, the process parameters table supplies all of the individual tasks and task durations involved in a particular unit operation. For example, the unit operation of inoculum preparation includes the individual tasks of setup, preincubation, incubation, and cleanup. Examples of unit operations for biopharmaceutical manufacturing and their associated operational parameters are shown in FIGs. 14-20.

A block flow diagram is generated at step 110 after unit operation list has obtained the operational parameters from the process parameters table at step 108. The block flow diagram illustrates each unit operation in the manufacturing process as a block with inputs for both incoming product and new material, as well as outputs for both processed product and waste. The block flow diagram is a simple yet convenient tool for quantifying material flows through the process in a way that allows the sizing of many key pieces of equipment relative to a given process scale.

The information in each block of the block flow diagram is generated from the parameters and sizing ratios from the process parameters table in the unit operation list, and block flow diagram calculation sets. A calculation set is a set of algebraic equations. The parameters and calculation sets are used to calculate the quantities of material inputs, product and waste outputs required for that unit operation based on the quantity of product material being received from the previous unit operation. Likewise, a given block flow diagram block calculates the quantity of product to be transferred to the next unit operation block in the manufacturing procedure. These calculations take into account the unit operation scheduling cycles identified at step 106, as further explained below.

A process time line is generated at step 112 after the block flow diagram is generated at step 110. The process time line is a very useful feature of the present invention. The process time line is generated from the unit operation list, the tasks associated with each of the unit operations, the scheduling cycles for each of the unit operations in the process, the process parameters from the master process parameters table and the volume of the material as calculated from the block flow diagram. The process time line is a relative time line in hours and minutes from the start date of the production process. The relative time is converted into days and hours to provide a time line for the beginning and ending times of each unit operation and its associated tasks for the entire biopharmaceutical drug production process.

The process time line is a very powerful tool for process design. The process time line can be used to accurately size pumps, filters and heat exchangers used in unit operations, by calculating the flow rate from the known transfer time and the volume of the material to be transferred, filtered or cooled. The process time line accurately predicts loads for labor, solution preparation, equipment cleaning, reagent, process utilities, preventative maintenance, quality control testing, etc.

FIG. 2 further illustrates step 102 of determining the necessary reactor vessel capacity. The amount of biopharmaceutical product to be produced in a given amount of time is determined in step 202. Normally, the amount of biopharmaceutical product required is expressed in terms of mass produced per year. The number of reactor vessel runs for a particular biopharmaceutical product per year is determined at step 204. Factors considered when determining the number of reactor vessel cycles for a particular biopharmaceutical product are, for example, the number of biopharmaceutical products produced in the reactor vessel (i.e., the reactor vessel is shared to produce different products), the reaction time for each cycle of the reactor vessel and the percentage of up-time for the reactor vessel over the year.

The yield of each batch or reactor cycle is calculated at step 206. The yield from each batch or a reactor cycle is process-dependent and is usually expressed in grams of crude product per liter of broth. Given the required amount of biopharmaceutical product per year from step 202, the number of reactor cycles available to produce the required biopharmaceutical product from step 204, and the yield of each reactor cycle from step 206, the necessary reactor volume to produce the required amount of biopharmaceutical product is calculated at step 208.

FIG. 3 illustrates a unit operation list for an exemplary microbial fermentation biopharmaceutical production process. The far left-hand column, column 302, lists the unit operation sequence numbers for each of the unit operations in the process. The exemplary microbial fermentation unit operation list includes 23 unit operations. The unit operation sequence number defines the order in which the unit operations occur. For example, unit operation sequence number 1, inoculum preparation, occurs first, before unit operation sequence number 2, flask growth. Column 304 shows the unit operation identifier codes associated with each of the unit operations in the unit operation list (see step 108). The unit operation identifier codes are used to bring operational parameters from the process parameters table into the unit operation list. For example, heat

exchange, unit operation list numbers 5, 8 and 10, has a unit operation identifier code 51.

As described above with reference to FIG. 1, after the unit operation sequence for a particular biopharmaceutical production process has been determined at step 104, the scheduling cycles associated with each unit operation is determined at step 106. Columns 306, 310 and 318 list the number of scheduling cycles for the microbial fermentation process of FIG. 3. Scheduling cycles are iterations of unit operations in the unit operation sequence, and occur in three levels. Additionally, each level of scheduling cycle has an associated offset duration that dictates the time period between the beginnings of successive scheduling cycles, shown in columns 308, 316 and 324. The latter two levels of scheduling cycles have an associated unit operation starting point and unit operation end point. That is, columns 312 and 314 specify the start and end unit operations, respectively, for Unit Operation Cluster Cycles, and columns 320 and 322 specify the start and end unit operations, respectively, for Batch Cycles.

Column 306 lists the number of Unit Operation Cycles for each of the unit operations in the microbial fermentation unit operation sequence. In the exemplary microbial fermentation unit operation sequence, each of the unit operations has only one cycle per unit operation. Again, Unit Operation Cycles define the number of iterations a unit operation is repeated in a process by itself before proceeding to the next unit operation.

Column 308 lists the cycle offset duration in hours for the Unit Operation Cycles. Since each of the unit operations in the microbial fermentation example of FIG. 3 has only one cycle per unit operation, there is no cycle offset duration for any of the unit operations. Cycle offset duration defines the time period between the beginnings of successive scheduling cycles.

Column 310 lists the Unit Operation Cluster Cycles for each of the unit operations in the microbial fermentation unit operation sequence. Unit operation sequence numbers 1-6 are defined as having three Unit Operation Cluster Cycles. Unit Operation Cluster Cycles defines the number of iterations a set of

consecutive unit operations are repeated as a group before proceeding to the next unit operation. In FIG. 3, for example, the set of unit operations 1-6, as defined in unit operation start column 312 and unit operation end column 314, cycle together as a group (e.g., the sequence of unit operations for the exemplary microbial fermentation process is 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6 and 7). Unit operations 1-6 cycle together as a group three times before the process continues to unit operation 7, as defined in column 310.

After unit operation sequence numbers 1-6 have cycled consecutively three times, the microbial fermentation production process continues at unit operation sequence number 7, resuspension of cell paste. After unit operation sequence number 7, the process continues with three Unit Operation Cluster Cycles of unit operation sequence numbers 8-10. The unit operations of heat exchange, cell disruption, and heat exchange are cycled consecutively three times, as defined in columns 310, 312 and 314. After unit operation sequence numbers 8-10 have cycled three times, the microbial fermentation production process continues at resuspension/surfactant, unit operation sequence number 11.

Unit operation sequence numbers 11 and 12 cycle together two times, as defined by columns 310, 312 and 314. After unit operation sequence numbers 11 and 12 have been cycled two times, the microbial fermentation production process continues without cycling from unit operation sequence number 13 through unit operation sequence number 23 to conclude the microbial fermentation production process.

Columns 326-332 of FIG. 3 represent the step wise recover (SWR) and overall recovery (OAR) percentages of the product and total proteins. SWR is the recovery of protein for the individual unit operation for which it is listed. OAR is the recovery of protein for the overall process up to and including the unit operation for which it is listed. The product recovery columns represent the recovery of the desired product protein from the solution in the process. The protein recovery columns represent the recovery of contaminant proteins from the solution which result in higher purity of the product solution.

FIG. 4 illustrates a unit operation list for an exemplary mammalian cell culture production process. Column 402 lists unit operation sequence numbers 1-19. Unit operation sequence numbers 1-19 define the order in which the unit operations of the mammalian cell culture production process occur. The most notable differences between the microbial fermentation process of FIG. 3 and the mammalian cell culture process of FIG. 4 are the multiple Unit Operation Cycles of unit operation sequence number 8 and the multiple Batch Cycles of unit operation sequence numbers 8-18.

Unit operation sequence number 8 of FIG. 4 illustrates the concept of multiple Unit Operation Cycles. Unit operation sequence number 8 is the unit operation of harvesting product rich growth media from and feeding fresh growth media into the mammalian cell reactor vessel. In most mammalian cell culture processes, the product is secreted by the cells into the surrounding growth media in the reactor vessel. To harvest the product, some of the product rich growth media is harvested from the reactor vessel to be processed to remove the product, and an equal amount of fresh growth media is fed into the reactor vessel to sustain production in the reactor vessel. The process of harvesting and feeding the reactor vessel can continue for many weeks for a single biopharmaceutical production process. Unit operation sequence number 8 is repeated seven times, or 7 Unit Operation Cycles (e.g., the unit operation sequence is 7, 8, 8, 8, 8, 8, 8, 8, 9). Note that the offset duration for unit operation sequence number 8 is 24 hours. The offset duration defines the time period between the Unit Operation Cycles. In the example of FIG. 4, unit operation sequence number 8 is repeated 7 times (7 Unit Operation Cycles) and each cycle is separated from the next by 24 hours, or one day. This corresponds to unit operation sequence number 8 having a duration of one week, with a harvest/feed step occurring each day.

FIG. 4 also illustrates the feature of multiple Batch Cycles. Batch Cycles is defined as the number of iterations a batch cycle is repeated in a given process that employs continuous or semi-continuous product synthesis. Each batch cycle results in a batch of product. A single biopharmaceutical production process,

therefore, may result in multiple batches of product. In the mammalian cell culture process example of FIG. 4, unit operation sequence numbers 8-18 are repeated together as a group eight times (column 418). Each of these cycles of unit operation sequence numbers 8-18 produce one batch of product (columns 420-422). The offset between each cycle of unit operation sequence numbers 8-18 is 168 hours, or one week (column 424).

In the example of FIG. 4, unit operation sequence numbers 8-18 proceed as follows: the reactor vessel is harvested and fed once each day for seven days; the results of the harvest/feed operation are pooled in unit operation sequence number 9 at the end of the seven days; unit operations 9-18 are then executed to process the pooled harvested growth media from unit operation sequence number 8. Unit operation sequence numbers 8-18 are cycled sequentially once each week to process an additional seven day batch of harvested growth media from unit operation sequence number 8. At the end of eight weeks, the mammalian cell culture process is completed.

FIG. 5 further illustrates step 108, cross referencing the unit operation sequence with the master process parameters table. The operational parameters in the process parameters table are those parameters necessary to simulate a particular unit operation. The parameters from the process parameters table define the key operational parameters and equipment sizing ratios for each unit operation in the unit operation sequence. The values for these parameters and ratios are variables which can be easily manipulated and ordered to model and evaluate alternative design scenarios for a given process scale. Examples of the process parameters associated with each unit operation are shown in FIGs. 14-20. It should be noted, however, that the list of unit operations, parameters, values, and scaling ratios is not exhaustive. One of ordinary skill in the art could expand the process parameters table to encompass additional unit operations and production processes for other batch process industries such as chemical pharmaceutical, specialty chemical, food, beverage, and cosmetics. Such

expansion would allow the present invention to simulate and schedule additional batch production processes for other such batch processes.

FIG. 5 illustrates the files necessary to cross-reference the unit operation list with the process parameters table in step 108. Exemplary unit operation list 502 for the biopharmaceutical production process and process parameters table 504 are input into processing step 506. Step 506 cross-references the unit operation list and process parameters table based on unit operation identification code (see FIG. 3). The parameters are copied from the process parameters table 504 into the unit operation list 502 to generate unit operation list 508.

FIG. 6 further illustrates exemplary process parameters table, 504. The operational parameters in the process parameters table are those parameters necessary to simulate a particular unit operation. The unit operation identification codes of process parameters table 504 are used in the cross-reference step 506 to assign the parameters from the process parameters table 504 to the unit operation list 502. Examples of operational parameters are the solutions involved in a particular unit operation, temperature, pressure, duration, agitation, scaling volume, etc. Additionally, the process parameters table defines all of the individual tasks and task durations involved in each unit operation. It should be noted, however, one of ordinary skill in the art could expand the process parameters table to encompass additional unit operations and production processes for other batch process industries such as chemical pharmaceutical, specialty chemical, food, beverage, and cosmetics. Such expansion would allow the present invention to simulate and schedule additional batch production processes for other such batch processes

FIG. 7 further illustrates step 110, generating a block flow diagram. A block flow diagram depicts each unit operation in the biopharmaceutical production process as a block with inputs for both incoming product and new material, as well as outputs for both processed product and waste. The material that flows through each of the unit operation blocks is quantified by calculation sets in each of the block flow diagram blocks. A unit operation block in a block

flow diagram is a graphical representation of a unit operation. A calculation set is a set of algebraic equations describing a unit operation. Some examples of outputs of the calculation sets are: required process materials for that unit operation, equipment performance specifications and process data outputs to be used for the next unit operation. Some examples of inputs to the calculation sets are: product quantity (mass) or volume (liters) from a previous unit operation, other parameters and/or multipliers derived from the process parameters table, as well as the design cycles defined in the unit operation list.

Block flow diagram 708 is generated from unit operation list 508 and block flow diagram calculation set 704. Block flow diagram calculation set 704 is an exhaustive list of unit operation identifier codes and the calculation sets associated with each unit operation identifier. Unit operation list 508 and block flow diagram calculation set 704 are linked together based on unit operation identifier code.

Step 706 calculates the block flow diagram material flow requirements and basic equipment sizing requirements from unit operation list 508 which includes all of the associated operational parameters from the process parameters table, and the block flow diagram calculation set 704. Block flow diagram 708 allows the sizing of many key pieces of equipment relative to a given process scale. Since the material flow quantities into and out of each unit operation is determined at step 706, the capacity of many equipment items involved in each unit operation can be determined. The block flow diagram also manages important information in the unit operation list 502 such as the percent recovery, percent purity and purification factor of the product in each unit operation. This information helps identify the steps in the process that may need optimization.

The following is an example calculation set for a tangential flow micro-filtration (TFMF) system unit operation. Tangential flow micro-filtration is an important process technology in biopharmaceutical manufacturing. This technology significantly extends the life of the filtration media and reduces the replacement cost of expensive filters.

TFMF generically requires the same steps to prepare the membrane for each use as well as for storage after use. The design parameters for each unit operation such as TFMF have been developed around these generic design requirements.

Generic Parameters (Variables) from the Process Parameters Table

Equipment Design Type	Plate & Frame
Membrane Porosity	0.2 micron
Membrane Flux rate	125 Liters/square meter/hour
Process Time	2 Hours
Retentate/Filtrate Rate	20 to 1
Flush Volume	21.5 Liters/square meter
Prime Volume	21.5 Liters/square meter
Wash Volume	0.5 % of Process Volume
Regenerate Volume	10.8 Liters/square meter
Storage Volume	21.5 Liters/square meter
% Recovery of Product	95%
% Recovery of Total Protein	80%
Clean In Place (CIP)	Yes
Steam In Place (CIP)	Yes

Input Values from Previous Unit Operation

Product Volume	1,000 Liters
Product Quantity	1.5 Kg
Total Protein Quantity	3.0 Kg

The calculation set for this unit operation first takes the incoming process volume and uses it as a basis of sizing the filtration membrane for the filtration system based on the above flux rate and required processing time:

1,000 Liters / 125 L/SM/Hr / 2 Hours = 4.0 SM of 0.2 micron membrane

After calculating the square meter (SM) of membrane required by this unit operation, the volumes of each of the support solutions can be calculated based on the above volume ratios:

Flush volume   21.5 Liters/SM x 4.0 SM = 86 Liters

Prime volume   21.5 Liters/SM x 4.0 SM = 86 Liters

Wash Volume 5 % of 1,000 Liters = 50 Liters

Regenerate   21.5 Liters/SM x 4.0 SM = 86 Liters

Storage   10.8 Liters /SM x 4.0 SM = 42 Liters

The flow rate of the filtrate is calculated from the volume to be filtered and the required process time:

1,000 liters / 2 hours = 8.3 liters/minute

The flow rate of the retentate is calculated based on the above retentate/filtrate ratio:

8.3 liters/minute x 20 = 167 liters/minute

Based on the input of the process volume to this unit operation and the above parameters, the equipment size, the filtration apparatus, the retentate pump, the support linkage and associated systems can be designed.

In addition, the input values for the quantity of product and contaminant protein received from the previous unit operation together with the recovery factors listed in the parameters allow the calculation of the cumulative recovery of product through this step, as well the percent purity of the product and the product purification factor for this step. This information is helpful for identifying steps in the manufacturing process which require optimization.

FIG. 8 illustrates an exemplary block flow diagram for the first five unit operations of the microbial fermentation process unit operation list of FIG. 3. Unit operations 1 through 5 are shown as blocks 802, 804, 806, 808 and 810. The input solutions to each of the steps are shown as arrows tagged with solution identifier information from the unit operation list 508. The process streams to which these solutions are added at each unit operation are also shown as arrows tagged with process stream identifier information. Working from the initial

process stream characteristics (P-101) in unit operation 1, inoculum prep, the volumes of input materials (solutions) and subsequent process streams in each of the unit operations is determined using scale-up ratios which are included in the information from the unit operation list 508 for each respective unit operation.

5 For example, the volume of solutions and process streams flowing into and out of each of unit operation blocks 802-810 in FIG. 8 is determined by the initial starting characteristics of the process stream P-101 and the volume of its associated input material S-101 in the first unit operation, block 802 and the scale up ratio in each of the successive unit operations, blocks 804-810. The solutions

10 involved in each of unit operation blocks 802-810 are likewise part of the information for each respective unit operation in the unit operation list 508.

FIG. 9 further illustrates step 112, generating the process time line. The process time line is generated (steps 904 and 906) from unit operation list 508 and block flow diagram calculation set 704. Unit operation list 508 contains

15 enough input information to generate a detailed process time line which includes the start and stop times for most of the tasks associated with each unit operation. The durations of some unit operation tasks are not scale dependent. The durations of other unit operation tasks are, however, scale dependent. In the latter case, as a process is scaled up, the amount of time required to complete a unit

20 operation task increases. In such cases, where duration of a unit operation task is scale dependent, block flow diagram calculation set 704 is required to calculate the quantity of material handled by the unit operation task. After the quantity of material handled by a unit operation task is determined, its duration can be determined. Examples of scale dependent task durations are the time required to

25 pump solutions from one storage tank to another, the amount of time required to heat or cool solutions in a heat exchanger, the amount of time required to filter product or contaminants from solution.

FIG. 10 is an example of a high-level process time line for a microbial fermentation process. The unit operation sequence of the process time line of

30 FIG. 10 corresponds to the unit operation list of FIG. 3. The high-level process

time line shown in FIG. 10 illustrates two Process Cycles of the microbial fermentation unit operation sequence, labeled "First Process Cycle" and "Second Process Cycle." A "Process Cycle" (PC) is a complete run of the biopharmaceutical production process, as defined by the unit operation sequence for the process.

The first two columns of the process time line of FIG. 10 identify the unit operation sequence number and unit operation description of the unit operation being performed, respectively. The first three sets of unit operations correspond to the three Unit Operation Cluster Cycles of unit operation sequence numbers 1-6 of FIG. 3. Three cycles of unit operations 1-6 are performed and the results are pooled into unit operation 7, pool harvests. The two columns to the right of the duration column identify the week and day that the particular unit operation is occurring in the first Process Cycle.

The day and the week each unit operation is performed is calculated from the start time of the process, as well as the cumulative duration of each of the previous unit operations. In the example of FIG. 10, Sunday is defined as the first day of the week. In the example of FIG. 10, the process sequence begins at unit operation 1, inoculum prep, on Friday of the first week. After unit operation 1 has completed (24 hours later, since unit operation 1 has a 24 hour duration) unit operation 2 is performed on Saturday. The begin and end times for each successive unit operation are calculated from the duration of the unit operation and end time of the previous unit operation. Note that FIG. 10 is calculated to the day and week only for the purposes of explanation. Usually the process time line is determined for each of the tasks associated with a unit operation to the minute.

As illustrated in FIG. 10, unit operation 7 occurs on Monday of the third week in the first Process Cycle. The third column from the left is the duration of each of the unit operations. After the three cycles of unit operations 1 through 6 have been pooled in unit operation 7, the process continues at unit operations 8 through 10, heat exchange, cell disruption and heat exchange. Each of unit operations 8 through 10 are cycled three times and the associated scheduling

information is contained in column to the right of the unit operation duration. Since each cycle of unit operations 8 through 10 have a duration of .5 hours, as shown in column 3, each cycle occurs on Monday of the third week in the process.

FIG. 11 illustrates the final unit operations of the process time line for the microbial fermentation process. After 3 cycles of unit operations 8 through 10 have been completed, unit operation sequence numbers 11 and 12 cycle together two times on Monday, week 3 of the first Process Cycle. After unit operation sequence numbers 11 and 12 have been cycled twice, the microbial fermentation production process continues without cycling from unit operation sequence number 13 through unit operation sequence number 22 to conclude the microbial fermentation production process. The durations and associated start times are listed for each of the unit operations 13-22.

FIGs. 12A-12H illustrate the preferred embodiment of a detailed process time line. The unit operation sequence of the process time line of FIGs. 12A-12H correspond to the unit operation list of FIG. 3. The process time line of FIGs. 12A-12H illustrates a single Process Cycle of the microbial fermentation unit operation sequence. The individual tasks associated with each unit operation are included after the unit operation. For example, in FIG. 12A, unit operation 1A, inoculum prep, consists of the individual tasks of set up, preincubation, incubation, and clean up. Columns 11-14 show the start date and time and finish date and time for each of the tasks in each unit operation. Since setup and clean up are not part of the critical path of the process, they do not directly affect the start and end times of following unit operations. The start and finish date and times for the set up and clean up operations of each of the unit operations are valuable because they ensure that the equipment will be available for each unit operation if the process time line is followed.

The process time line of FIGs. 12A-12H includes examples of unit operation task duration calculations. Row 20, column 15 of FIG. 12A, which corresponds to the harvest task of unit operation 3A, seed fermentation, is an

example of a duration calculation. As stated above, the duration of some unit operations is process scale dependent (i.e., the duration is dependent upon the volume processed). The harvest task in the seed fermentation unit operation is an example of a task whose duration is process scale dependent. In column 15, the calculations column, information listed for the harvest task is 50 liters, 1.7 liters/minute (LPM), and 0.5 hours. Fifty liters represents the volume of material that is harvested during a harvest task. 1.7 liters/minute represents the rate at which the solution is harvested. Given the volume to be harvested and the flow rate of the harvest, the duration of the harvest task is calculated to be 0.5 hours. Each task in a unit operation that is volume dependent has its duration calculated in order to generate the process time line of FIGs 12A-12H.

The process time line of FIGs. 12A-12H can be resolved to minutes and seconds, if necessary. The accuracy of the process time line allows the precise planning and scheduling of many aspects of the batch manufacturing process. The process time line scheduling information can be used to schedule manufacturing resources such as labor, reagents, reusables, disposables, etc., required directly by the manufacturing process. Pre-process support activities such as solution preparation, and equipment prep and sterilization, required to support the core process, including the labor, reagents, etc. can be scheduled, cost forecasted and provided for. Post-process support activities such as product formulation, aseptic fill, freeze drying, vial capping, vial labeling and packaging required to ship the purified product in a form ready for use may be added to the process time line and managed. Based on the process time line, labor, reagents, etc., required to support these post-process support functions can be acquired and managed. One of the most important aspects of the present invention is the determination of process utility loads such as USP Purified Water, Water For Injection, Pure Steam, etc., for all of the manufacturing equipment. The process time line can be used to determine the peak utility loading, and utility requirements for the facility. Building utility loads such as building steam, heating, ventilation, air conditioning, plumbing, etc., for all manufacturing

equipment, process areas and facility equipment can be determined based on the process time line and the equipment associated with each of the unit operations. The process time line can be used to measure the time that the equipment has been in service to schedule preventative maintenance of all plant equipment, Quality Assurance activities including instrument calibration, automated batch documentation, etc. and Quality Control activities including process system maintenance, raw material testing, in process testing and final product testing, etc.

In an alternative embodiment of the present invention, a refinement can be made to the generation of a process time line (PTL), as shown in FIG. 9 (step 112 of FIG. 1), based on the three levels of scheduling cycles--"Unit Operation Cycles," "Unit Operation Cluster Cycles," and "Batch Cycles"--discussed above.

The refinement (i.e., new mechanism) focuses on how to apply the offsets associated with each design cycle to the PTL. In cases where the offset for any of the levels of scheduling cycles is equal to zero, as soon as the cycle is completed it immediately begins the next cycle at that same cycle level, assuming another cycle has been specified. Otherwise, the scheduling mechanism passes to the next unit operation specified and its respective set of cycle conditions. This new mechanism clarifies the impact an offset of greater than zero for any of the scheduling cycle levels has on the PTL.

To implement this new mechanism, a unit operation tagging convention based on a four-field delimited string is used to identify each cycle iteration of a unit operation module in a PTL (PTML) by its respective cycle level and cycle count. A PTML is a section of the PTL having a set of tasks with respective scheduling calculations that are associated with a given unit operation. The above delimited fields are defined as follows:

DELIMITED FIELD	VALUE
Field 1	Unit Operation Sequence Number
Field 2	Batch Cycle Iteration
Field 3	Cluster Cycle Iteration

DELIMITED FIELD	VALUE
Field 4	Unit Operation Cycle

Together, the four-fields of the delimited string makes up a Delimited String Code (DSC) (e.g., "3.2.1.1") that provides a unique identifier for each PTLM in the PTL. The DSCs are based upon the above-described sequence and design cycle information for each design cycle iteration for each unit operation in the PTL.

As discussed above, a PTL consists of unit operations decomposed into unit operation tasks that are required to complete a unit operation procedure. These tasks can be divided in to three basic categories: Unit Operation Set Up (i.e., equipment preparation), Unit Operation Execution (i.e., equipment usage), and Unit Operation Clean Up (i.e., equipment clean up). Tasks associated with Unit Operation Execution can be said to be on the critical path of the PTL and usually involve manipulation of the product in preparation for the next unit operation in the process procedure.

Unit Operation Set Up Tasks (SUTs) are performed before Unit Operation Execution Tasks in order to prepare equipment for the Unit Operation Tasks that are on the critical path of the PTL. Generally, SUTs can be performed any time before the first Unit Operation Execution task is scheduled to be started based on the availability of the product from the previous unit operation. However, SUTs cannot be back scheduled to the extent that they interfere with completion of the previous usage cycle for the respective set of equipment or compromise the integrity of the process or product in any way. Unit Operation Clean Up Tasks (CUTs) are required in order to clean and prepare equipment for its next use. The above definitions of the three basic tasks are important relevant to accurately defining the impact of the above design cycles and their associated offsets on task scheduling in the PTL.

An application of the offset refinement can be explained, for example, with reference to mammalian cell culture production process (see FIG. 4

described above) and microbial fermentation process (See FIG. 5 described above). Referring to FIG. 21, an example showing the harvest and initial purification steps associated with a mammalian cell culture process is presented. A mammalian cell culture process provides a good example of the use of Batch Cycles as well as Unit Operation Cycles and their respective offsets. It is based on a batch process that involves continuous or semi-continuous product synthesis with batch purification of the crude product produced by the cell culture reactor.

Because a mammalian cell culture reactor may be in production for 60-90 days, the product enriched media that the cells grow in is typically harvested either continuously or in batch harvest cycles every 1, 2 or 3 days. In each harvest cycle a fresh supply of media is fed to the reactors as the product enriched media is harvested. Thus, the mammalian cell process illustrated in FIG. 21 demonstrates how Batch Cycles and their offsets impact the generation of a consecutive PTL together with the Unit Operation Cycles and their offsets.

Referring to FIG. 22 an example showing the initial purification steps associated with a microbial fermentation process is presented. A microbial fermentation process provides a good example of the use of two types of Cluster Cycles and their respective offsets and how they and their offsets impact the generation of a consecutive PTL.

Referring again to FIG. 21, where media is batch harvested every 24 hours for 3 UC's (Unit Operation 1) and these daily harvests are pooled every 72 hours to be batch purified (Unit Operations 2-4), a base model which demonstrates the use of offsets for Batch Cycles (BC) as well as Unit Operation Cycles is present. In this case a BC is defined by unit operations 2-4 as a subset of the process cycle defined by unit operations 1-4. The Batch Cycle Offset (BCO) in this case is 72 (3 days). Every 3 days a new batch of material is begun. In a PTL for this example, if one were to demonstrate a consecutive time line with iterative Batch Cycles, the start of the First Critical Path Task (FCPT) would have to be offset from the beginning of the FCPT from the previous BC.

Referring again to FIG. 22, where a cluster of three concurrent unit operations is cycled 3 times in order to "disrupt" the cells harvested from a fermentor, a concurrent or simultaneous cluster cycle is presented (Unit Operations 10305-10307). In a simultaneous cluster cycle all the unit operations in a given cluster cycle operate simultaneously versus sequentially during each cluster cycle. In this case the inlet heat exchanger, cell disruption and outlet heat exchanger all operate simultaneously to each other during each Cluster Cycle. In Unit Operations 10309-10310, a sequential cluster cycle is presented where the first unit operation in the cluster completes is unit operation cycle before the next unit operation in the cluster cycle begins and so on during each cluster cycle.

Referring to FIGs. 23A-E, a PTL generated based on the design cycles defined in the Unit Operation List in FIG. 21 is shown. In this PTL, there is a PTL Module (PTLM) for each unit operation cycle required to generate a consecutive time line based on the design cycle patterns defined in the Unit Operation List. Column 2302 lists the DSC for each PTLM in the sample PTL. Column 2304 lists calculations required to determine the duration of each of the calculated tasks associated with a unit operation. Column 2306 lists the duration of each task as determined by either calculation or manual entry. Column 2308 lists schedule adjustment factors (SAF) to forward schedule or back schedule a task duration. Column 2310 lists the adjusted duration of a task based on the calculated duration in column 2306 and the adjustment factor in column 2308.

Column 2312 lists the finish times, in hours, for each of the SUTs. These finish times are back calculated from the FCPS time based on the task duration and SAF for each SUT (as further explained below). Column 2314 lists the finish times for each of the Critical Path Tasks (CPT) associated with a unit operation. The finish times are forward calculated from the First Critical Path Task End Time (FCTPE) based on the task duration and SAF for each respective CPT (further explained below). Column 2316 lists the finish times for the CUTs associated with a unit operation. Finish times for these CUTs are calculated from the finish time of the Critical Path End Time for the Current Unit Operation

(CCPE) and the adjusted durations for each of the subsequent CUTs. Column 2318 lists the calculated start time for each task in the PTL based upon the end time for the respective task in columns 2312, 2314, or 2316, and their respective task duration and SAFs from columns 2306 and 2308, respectively. Column 2320 lists the calculated end time for each task in the PTL based on the end times determined in columns 2312, 2314 or 2316.

Referring to FIG. 24, a high level flow chart that illustrates the method for determining the impact of design cycle offsets on the Current Critical Path Start Time (CCPS) for a respective Unit Operation Cycle in the PTL is shown. In Step 2402, the Batch Cycle Iteration Field (BCIF) for the PTLM (DCS Field 2) is evaluated to determine if the current unit operation sequence number is the start unit operation for a Batch Cycle Iteration as defined in column 2120 in FIG. 21. If it is, the CCPS time for the new Batch Cycle Iteration is calculated from the start time from the previous BC plus the BCO. If not, the BC test is ended and the cluster cycle conditions for the current unit operation are then evaluated in Step 2404.

Referring to FIG. 25, the evaluation in step 2402 is further illustrated. The BCIF for the Current Unit Operation in Step 2502 is evaluated in step 2504. If the BCIF for the Current Unit Operation is not greater than one, the BC test is ended (Step 2506) and the Cluster Cycle Test is performed. If the BCIF is greater than one, the Batch Cycle Start Unit Operation (BCSO) is obtained from column 2122 from FIG. 21 (Step 2508). The BCIF is evaluated in Step 2510 to determine if it is equal to the BCSO. If the BCIF is not equal to the BCSO, the Batch Cycle Test is ended in Step 2506 and the Cluster Cycle Test is performed. If the BCIF is equal to the BCSO, then the BCO is obtained from Column 2126 in FIG. 21 (Step 2512). In Step 2516, the BSO is evaluated to determine if it is greater than zero. If the BCO for the Current Unit Operation is not greater than zero, the Batch Cycle Test is ended in Step 2506 and the Cluster Cycle Test is Performed. Otherwise, the CCPS is set equal to the PCPS plus the BCO (Step 2518). The resulting value from FIG. 25 is then stored for further evaluation in Step 2408.

A Cluster Cycle occurs when a set of two or more Unit Operations cycle together prior to the next steps in the manufacturing procedure within a BC. In some cases the Unit Operations in the Cluster Cycle follow each other sequentially where the CCPS of a subsequent Unit Operation in a CC follows the End Time for the Last Critical Path Task of the previous Unit Operation (PCPE) in the UC. An example of this type of CC is found in FIG. 22 in a sample case of the initial purification stage of a Microbial Fermentation Process. In this sample case, the Inclusion Body Wash steps in Unit Operations 10309 - 10310 are repeated twice before continuing with Unit Operation 10311. The Dilution Step is competed before the Centrifugation Step is started. After Unit Operation 10310 is completed Unit Operations 10309 - 10310 are repeated a second time in the same sequential manner before continuing with the next unit operation 10311, thus completing the second cluster cycle. In other CC cases the Unit Operations subsequent to the first Unit Operation in a CC occur concurrently to the first Unit Operation in the CC. In this latter case the CCPS of each Unit Operation in the CC that is subsequent to the first Unit Operation in the CC is set equal to the Critical Path Start Time of the First Unit Operation in the CC. An example of this type of CC is also found in FIG. 22. More specifically, the Cell Disruption Steps in Unit Operations 10305 - 10307 are repeated three times before continuing with Unit Operation 10308. In this case Unit Operations 10306 and 10307 each occur simultaneously to the first (10305). In other words, all three Unit Operations in the Cluster Cycle share the same CCPS.

Step 2404 of FIG. 24 illustrates a CC test being performed to determine if the current Unit Operation in the PTL is the start of a new CC. In Step 2404, the Cluster Cycle Iteration Field (CCIF) for the current unit operation (DCS Field 3) is evaluated to determine if the Current Unit Operation sequence number in column 2202 is the start unit operation for a Cluster Cycle Iteration as defined in column 2214 in FIG. 22. If it is, the CCPS time for the new Cluster Cycle Iteration is calculated from the start time from the previous CC plus the Cluster Cycle Offset (CCO) from column 2218 as further explained below.

Referring to FIG. 26, the CC test in step 2404 is further illustrated. The CCIF for the current Unit Operation in Step 2602 is evaluated in Step 2604. If the CCIF is not greater than one, the CC test is ended in step 2606 and the Unit Operation Cycle Test is performed as in Step 2406. If the CCIF is greater than

5   one, the Cluster Cycle Start Unit Operation (CCSO) is obtained from column 2214 in FIG. 22 (Step 2608). In Step 2610, the CCIF is evaluated to determine if it is equal to the CCSO. If the CCIF is not equal to the CSCO the CC test is ended in Step 2606 and the Unit Operation Cycle Test is performed. If the CCIF is equal to the CSCO, then CCO for the Current Unit Operation is obtained from

10   column 2218 (Step 2612). In step 2614 the CCO is evaluated to determine if it is greater than zero.

If the CCO is greater than zero, CCPS is set equal to Critical Path Start for the Previous Cluster Cycle plus the CCO (Step 2616). If the CCO is not greater than zero, the CCO is evaluated in Step 2618 to determine if it is equal to

15   "S" meaning that the current Unit Operation is to start simultaneously (concurrently) to the first Unit Operation in the current Cluster Cycle. If the CCO is not equal to "S", the CC test is ended in Step 2606 and Unit Operation Cycle Test is performed. Otherwise, the CCPS is set equal to the Critical Path Start for the Previous Cluster Cycle (Step 2620). The resulting value from FIG. 26 is

20   stored for further evaluation in Step 2408.

Step 2406 of FIG. 24 illustrates a UC test being performed after a CC test. In Step 2406, the Unit Operation Cycle Iteration Field (UCIF) for the current unit operation (DCS Field 4) is evaluated to determine if it is greater than one. If it is not, the Unit Operation Cycle Test ends. If it is, the CCPS time for the new

25   Unit Operation Cycle Iteration is calculated from the start time from the previous UC plus the Unit Operation Cycle Offset (UCO) as further defined below. Otherwise the CCPS time for the current Unit Operation continues from the PCPE of the previous Unit Operation.

Referring to FIG. 27, the evaluation in step 2406 is further illustrated. In

30   Step 2702, the UCIF for the current Unit Operation is obtained. In Step 2704 it

is evaluated to determine if it is greater than one. If the UCIF is not greater than one, the Cycle Offset Test ends as indicated by Step 2706. If the UCIF is greater than one, then the UCO for the Current Unit Operation in column 2107 is obtained (step 2708). In step 2710, the UCO is evaluated to determine if it is

5   greater than zero. If the UCO is greater than zero, CCPS is set equal to Critical Path Start for the Previous Unit Operation Cycle plus the UCO (Step 2712). If the UCO is not greater than zero, the UCO is evaluated in Step 2714 to determine if it is equal to "S" meaning that the current Unit Operation is to start simultaneously (concurrently) to the previous Unit Operation. If UCO is not

10   equal to "S", the Unit Operation Cycle Test is ended (Step 2706). Otherwise, the CCPS is set equal to PCPS (Step 2716). The resulting value from FIG. 27 is stored for further evaluation in Step 2408.

         Step 2408 of FIG. 24 evaluates the results of the above three Cycle Offset Tests do see if there is a positive result from any of them, indicating that the

15   current unit operation is the beginning of an iteration of one of the three levels of design cycles to which a design cycle offset applies. If there is not a positive result, CCPS is set equal to PCPE (Step 2410). If it there is a positive result CCPS is set equal to the latest start time produced by the three tests (Step 2412).

         FIG. 28 further illustrates Steps 2408-2412 of FIG. 24. In Steps 2802,

20   2804 and 2806 the Design Cycle Test Results from the Batch Cycle Test, Cluster Cycle Test and Unit Operation Cycle Test are obtained, respectively. In Step 2808, these obtained values are evaluated to find the maximum CCPS value (MCCPS). In step 2810, the MCCPS value is evaluated to determine if it is greater than zero. A MCCPS value equal to zero in Step 2808 indicates that the

25   current unit operation is not a start unit operation for any design cycle iteration to which a design cycle offset applies. Therefore the CCPS time for the current unit operation in this case is PCPE (Step 2812). A MCCPS value greater than zero indicates that the current unit operation is the start unit operation for at least one design cycle iteration to which an offset greater than zero applies at the

30   respective design cycle level. Therefore the CCPS time for the current unit

operation in this case is equal to MCCPS from Step 2808. Note that the Design Cycle Tests illustrated in FIGs. 24-31 only impact the PTL if the Current Unit Operation (CUO) is the Start Unit Operation (SUO) for an iteration of at least one of the above design cycles and the offset of that design cycle is greater than zero or equal to "S". Otherwise the CCPS starts at PCPE even if the CUO is the SUO for a given Design Cycle.

Once the accurate CCPS has been determined via the above procedure, the CCPS times for each of the Design Cycle levels needs to be updated to be referenced by the Design Cycle Check mechanism for the next PTLM in the PTL as shown in Step 2414 of FIG. 24. This update mechanism is dependent on the current PTLM being the SUO of a respective Design Cycle set and the number of cycles for a respective Design Cycle level being greater than one. This update mechanism does not depend on there being an Offset Value greater than zero associated with the respective Design Cycle, as does the above Design Cycle Check mechanism.

FIG. 29 further illustrates the procedure for updating the CCPS for the current Batch Cycle for the current PTLM. The BCIF is obtained in Step 2902 and evaluated in Step 2904. If the BCIF is not greater than one, the current PTLM is not the start of a new Batch Cycle because there are no reiterating Batch Cycles defined for the current PTL in the Unit Operations List (FIG. 21, Column 2120). Therefore, the Start Time for the CCPS for the current Batch Cycle (CBCS) is set equal to the CBCS for the previous Unit Operation (Step 2906). If BCIF is greater than one, the BCIF is further evaluated in Step 2908 to determine if it is equal to the BCSO. If the BCIF is not equal to the BCSO, the current Unit Operation is not the Start Unit Operation for a new iteration of a Batch Cycle. Therefore, the CBCS is set equal to CBCS for the Previous Unit Operation (Step 2906). If the BCIPF is equal to the BCSO, the current Unit Operation is the Start Unit Operation for a new iteration of a Batch Cycle and CBCS is set equal to the CCPS for the Current PTLM as determined by the procedures illustrated in FIGs. 25 - 28.

FIG. 30 further illustrates the procedure for updating the CCPS for the current Cluster Cycle for the current PTLM. The CCIF is obtained in Step 3002 and evaluated in Step 3004. If the CCIF is not greater than one, the current PTLM is not the start of a new Cluster Cycle iteration because there are no reiterating Cluster Cycles defined for the current PTL in the Unit Operations List (FIG. 21, Column 2112). Therefore, the Start Time for the CCPS for the current Cluster Cycle (CCCS) is set equal to the CCCS for the previous Unit Operation (Step 3006). If CCIF is greater than one, the CCIF is further evaluated in Step 3008 to determine if it is equal to the CCSO. If the CCIF is not equal to the CCSO, the current Unit Operation is not the Start Unit Operation for a new iteration of a Cluster Cycle. Therefore, the CCCS is set equal to CCCS for the Previous Unit Operation (Step 3006). If the BCIPF is equal to the CCSO, the current Unit Operation is the Start Unit Operation for a new iteration of a Cluster Cycle and CCCS is set equal to the CCPS for the Current PTLM as determined by the procedures illustrated in FIGs. 25 - 28.

FIG. 31 further illustrates the procedure for updating the CCPS for the current Unit Operation Cycle for the current PTLM. The UCIF is obtained in Step 3102 and evaluated in Step 3104. If the UCIF is not greater than one, the current PTLM is not the start of a new Unit Operations Cycle iteration because there are no reiterating Unit Operations Cycles defined for the current PTL in the Unit Operations List (FIG. 21, Column 2112). Therefore, the Start Time for the CCPS for the current Unit Operations Cycle (UCCS) is set equal to the UCCS for the previous Unit Operation (Step 3106). If UCIF is greater than one, the CCCS is set equal to the CCPS for the Current PTLM as determined by the procedures illustrated in FIGs. 25 - 28.

After the accurate CCPS has been determined for a PTLM and the CCPS for each of the Design Cycle Levels has been updated, the Task Start/Stop Times for each Critical Path Task (CPT) for the Current Unit Operation is calculated as illustrated in Step 2416. FIG. 32 further illustrates the procedure for determining the Task Start/Stop Times for each CPT for the Current Unit Operation. In Step

3202 the CCPS is obtained for the current PTLM via the procedure illustrated in Steps 2402-2412. In Step 3204 the duration of the First CPT (FCPT) is obtained from the calculations illustrated in FIG. 23, Column 2304. In Step 3206 the SAF is entered manually by the user to provide the user a means of delaying (or

5   starting earlier) the start of a CPT. The SAF can be used to manually adjust the PTL in order to start and end tasks at more favorable times with respect to shift scheduling of labor resources and equipment utilization. For example, if a task is scheduled by automatic PTL calculation to end during a night shift when no staffing is present, and staffing is required to end the task and/or start the

10   following task on the PTL, the SAF can be used to delay the start of a Critical Path Task such that the end of the task occurs when staffing will be present as required. The use of the SAF, as well as its limits of use, is further discussed below.

         In step 3208 the CCPS for the current PTLM is added to the duration and

15   the SAF of the FCPT to determine the end time of the FCPT (FCPTE). The present embodiment focuses on the determination of task end times from which the start time is back calculated. Alternatively, the FCPT Start time can be determined first by adding the SAF to the CCPS for the current PTL. In this latter case the FCPTE is calculated by adding the duration for the FCPT to this start

20   time. The former method was chosen for this example because the end time of a CPT task is generally the focus of attention when optimizing a PTL. Generally, the SAF for all Critical Path Tasks is usually greater than or equal to zero. If the SAF for a CPT is less than zero, the start time for the CPT may conflict with the end time for the previous CPT.

25   For example, if the transfer of a product from a "product hold vessel" to a "product mix vessel" is defined as CPT 1, and the mixing of the product in the mix vessel is defined as CPT 2, the start/end times for CPT 2 cannot be back scheduled such that the start time for CPT 2 begins before the transfer of the product from CPT 1 is complete. However, there may be instances when it is

preferable for the start time for CPT2 to begin before CPT 1 is completed. Such an example may involve the cooling or heating of product in a product mix vessel associated with CPT 2. If the start of the cooling or heating cycle for the product is associated with CPT 2, the heating or cooling cycle may need to be started after the product transfer has started and before the product transfer is complete. In this case the SAF can be a negative value. However, in general, the SAF is used to delay the start/end of a CPT from the completion time of the previous CPT, provided the delay does not conflict with use of the unit operation resources for another scheduled event or comprise the process or product in any other way.

Returning to FIG. 32, in Step 3214, the end time for a subsequent CPT is similarly determined by adding the duration (Step 3210) and SAF (Step 3212) for the subsequent CPT to the end time from the previous CPT for a given PTLM. The start time for the subsequent CPT can then be back calculated from the end time by subtracting the sum of the duration and SAF for the given subsequent task. As in the case of the FCPT, the start time for the subsequent CPT can be alternatively calculated first by adding the SAF to the end time for the previous CPT within the PTLM. In this case, the end time for the subsequent CPT is then calculated by adding the duration for the subsequent CPT to this sum. The procedure for determining the Start/Stop time for a subsequent CPT in Steps 3210 to 3214 is repeated for each subsequent CPTs associated with a given PTLM, thereby generating a critical path time line for the current PTLM.

Referring to FIG. 33, the procedure for determining the Task Start/Stop Times for each SUT for the Current Unit Operation is further illustrated. The FCPTE for the Current PTLM is obtained from Step 3210 for the current PTLM. In Step 3302, the duration of the Last SUT (SUTL) is obtained from the calculations illustrated in FIG. 23, Column 2304. In some cases, the duration of SUTL will be entered manually by the user rather than calculated. In Step 3304, the Forward/Batch Schedule Adjustment Factor (SAF) for the SUTL is entered manually to provide the user a means of starting the SUT earlier. In most cases the SAF for all SUTs will be less than or equal to zero, implying that the SUT

will start in time to be completed before the next task in the PTL begins or will start earlier than the minimum start time. If the SAF for a CPT greater than zero, the start time for the CPT may conflict with the end time for the previous CPT.

In step 3306, the duration of the FCPTE for the current PTLM is subtracted from the FCPTE and added to the SAF for the SUTL to determine the end time of the SUTL. As previously stated, the present embodiment focuses on the determination of task end times from which the start time is back calculated. Alternatively, the SUTL Start Time can be determined by subtracting the duration of the FCPTE and SUTL from the FCPTE and adding the SAF to the result. In this latter case, the end time for the SUTL is calculated by adding the duration for the SUTL and SAF to its start time. The former method was chosen for this example because the end time of a SUT task is generally the focus of attention when optimizing a PTL.

In Step 3312, the end time for a previous SUT is similarly determined by subtracting the duration of the current SUT from (Step 3308) from the end time for the current SUT and adding the result to the SAF (Step 3310) for the previous SUT. The start time for the previous SUT can then be back calculated from its end time by subtracting its duration from its end time. As in the case of the SUTL, the start time for the previous SUT can be alternatively calculated first by subtracting the duration of the current SUT and previous SUT from the end time of the current SUT and adding the SAF for the previous SUT to the result. In this case, the end time for the previous SUT is then calculated by adding the duration for the previous SUT to this sum. The procedure for determining the Start/Stop time for a previous SUT in Steps 3308 to 3312 is repeated for each previous SUT associated with a given PTLM, thereby generating a critical path time line for the current PTLM.

Referring to FIG. 34, the procedure for determining the Task Start/Stop Times for each CUT for the Current Unit Operation is further illustrated. The CCPE for the Current PTLM is obtained in Step 3402. In Step 3404, the duration of the First CUT (CUTF) is obtained from the calculations illustrated in FIG. 23,

Column 2304. In some cases the duration of CUTF will be entered manually by the user rather than calculated. In Step 3406, the SAF for the CUTF can be altered manually from its default of zero to provide the user a means of delaying the start of the CUT. In most cases the SAF for all CUTs will be greater than or equal to zero. This implies that the CUT will start when the last CPT for a given PTLM is completed (SAF for CUT is set equal to zero) or sometime after CCPE (SAF for CUT is greater than zero) provided that all CUTs associated with a PTLM are completed before the next use cycle of the equipment and resources associated with the current PTLM. If the SAF for a CPT is less than zero, the start time for the CPT may conflict with the end time for the last CPT (or previous CUT). In step 3408, the CCPE for the current PTLM obtained in Step 3402 is added to the duration and SAF for the CUTF to determine the end time of the CUTF. As previously stated, the present embodiment focuses on the determination of task end times from which the start time is back calculated. Alternatively, the CUTF Start Time can be determined first by adding the SAF to the CCPE. In this latter case the end time for the CUTF is calculated by adding the duration for the CUTF to its calculated start time. The former method was chosen for this example because the end time of CUT task is generally the focus of attention when optimizing a PTL.

In Step 3414, the end time for a subsequent CUT (CUTS) is similarly determined by adding the duration (Step 3410) and SAF (Step 3412) for the CUTS to the end time from the current CUT for a given PTLM. The start time for the CUTS can then be back calculated from the end time by subtracting the duration of the CUTS from its end time. As in the case of the CUTL, the start time for the CUTS can be alternatively calculated first by adding the SAF to the end time for the current CUT within the PTLM. In this latter case, the end time for the CUTS is then calculated by adding the duration for the previous CUT to this sum. The procedure for determining the Start/Stop time for a CUTS in Steps 3408 to 3412 is repeated for each CUTS associated with a given PTLM, thereby generating a critical path time line for the current PTLM.

The Process Start Time (PST) for a PTL is entered manually by the user and provides the start point for all of the above time line calculations. The first PTLM in a PTL takes the PST as the CCPS to begin the PTL. Once the CCPS for the first PTLM has been entered, it is possible to calculate start and end times of all the CPTs, as well as the SUTs and CUTs associated with the first PTL. Subsequently, the start and end times of the CPTs, SUTs and CUTs for all subsequent PTLMs can be determined based on the above procedure.

The present invention may be implemented using hardware, software or a combination thereof and may be implemented in a computer system or other processing system. In fact, in one embodiment, the invention is directed toward a computer system capable of carrying out the functionality described herein. An example computer system 1301 is shown in FIG. 13. The computer system 1301 includes one or more processors, such as processor 1304. The processor 1304 is connected to a communication bus 1302. Various software embodiments are described in terms of this example computer system. After reading this description, it will become apparent to a person skilled in the relevant art how to implement the invention using other computer systems and/or computer architectures.

Computer system 1302 also includes a main memory 1306, preferably random access memory (RAM), and can also include a secondary memory 1308. The secondary memory 1308 can include, for example, a hard disk drive 1310 and/or a removable storage drive 1312, representing a floppy disk drive, a magnetic tape drive, an optical disk drive, etc. The removable storage drive 1312 reads from and/or writes to a removable storage unit 1314 in a well known manner. Removable storage unit 1314, represents a floppy disk, magnetic tape, optical disk, etc. which is read by and written to by removable storage drive 1312. As will be appreciated, the removable storage unit 1314 includes a computer usable storage medium having stored therein computer software and/or data.

In alternative embodiments, secondary memory 1308 may include other similar means for allowing computer programs or other instructions to be loaded

into computer system 1301. Such means can include, for example, a removable storage unit 1322 and an interface 1320. Examples of such can include a program cartridge and cartridge interface (such as that found in video game devices), a removable memory chip (such as an EPROM, or PROM) and associated socket, and other removable storage units 1322 and interfaces 1320 which allow software and data to be transferred from the removable storage unit 1322 to computer system 1301.

Computer system 1301 can also include a communications interface 1324. Communications interface 1324 allows software and data to be transferred between computer system 1301 and external devices. Examples of communications interface 1324 can include a modem, a network interface (such as an Ethernet card), a communications port, a PCMCIA slot and card, etc. Software and data transferred via communications interface 1324 are in the form of signals which can be electronic, electromagnetic, optical or other signals capable of being received by communications interface 1324. These signals 1326 are provided to communications interface via a channel 1328. This channel 1328 carries signals 1326 and can be implemented using wire or cable, fiber optics, a phone line, a cellular phone link, an RF link and other communications channels.

In this document, the terms "computer program medium" and "computer usable medium" are used to generally refer to media such as removable storage device 1312, a hard disk installed in hard disk drive 1310, and signals 1326. These computer program products are means for providing software to computer system 1301.

Computer programs (also called computer control logic) are stored in main memory and/or secondary memory 1308. Computer programs can also be received via communications interface 1324. Such computer programs, when executed, enable the computer system 1301 to perform the features of the present invention as discussed herein. In particular, the computer programs, when executed, enable the processor 1304 to perform the features of the present

5

10

15

20

25

invention. Accordingly, such computer programs represent controllers of the computer system 1301.

In an embodiment where the invention is implemented using software, the software may be stored in a computer program product and loaded into computer system 1301 using removable storage drive 1312, hard drive 1310 or communications interface 1324. The control logic (software), when executed by the processor 1304, causes the processor 1304 to perform the functions of the invention as described herein.

In another embodiment, the invention is implemented primarily in hardware using, for example, hardware components such as application specific integrated circuits (ASICs). Implementation of the hardware state machine so as to perform the functions described herein will be apparent to persons skilled in the relevant art(s).

In yet another embodiment, the invention is implemented using a combination of both hardware and software.

While the invention has been particularly shown and described with reference to preferred embodiments thereof, it will be understood by those skilled in the relevant art that various changes in form and details may be made therein without departing from the spirit and scope of the invention.